Paper

The Use of Genetic Algorithms for Searching Parameter Space in Gaussian Process Modeling

Agnieszka Krok

Tadeusz Kościuszko Cracow University of Technology, Cracow, Poland

Abstract— The aim of the paper is to present the possibilities of modeling the experimental data by Gaussian processes. Genetic algorithms are used for finding the Gaussian process parameters. Comparison of data modeling accuracy is made according to neural networks learned by Kalman filtering. Concrete hysteresis loops obtained by the experiment of cyclic loading are considered as the real data time series.

Keywords—Gaussian processes, genetic algorithms.

1. Modeling Time Series

Modeling processes that are time-dependent is made based on many techniques. Methods for time series analyzes may be: correlation analyzes, autoregressive or moving average model, trend estimation and decomposition of time series, principal component analysis, Fast Fourier Transform, continuous wavelet transform. Also many tools are used for time series modeling, e.g., general state space models, unobserved components models, and machine learning methods such as artificial neural networks (ANN), support vector machines, Gaussian processes (GPs) [1], [2]. The main concern of the paper is Gaussian processes modeling. Genetic algorithms (GAs) are well known tools for searching the space of sub-optimal solutions, for example for supporting neural networks learning processes [3]. The investigation was made for the possibility of transferring the techniques known in the field of neural networks for accelerating the search of Gaussian models parameters.

2. Motivation and Related Background

Nowadays Gaussian processes are used for modeling different kind of data and wide variety of time series. In [4] GPs were used for modeling super resolution images, confirming the ability to deal with data read out from the image. In [5] complicated problem of probabilistic prediction of Alzheimer's disease from multimodal image was solved. GPs were also used for modeling time dependent processes inside materials, i.e. wax precipitation model in crude oil systems [6].

Also time depended signals such as speech [7], wind energy systems [8], and facial expressions [8], economical time series, was successfully modeled using GPs.

Modeling stress-strain hysteresis loops involves the representation of changes in time the material properties during tension-compression test. The data for analysis were discrete points taken from the graphical representation of stress-strain relation considered as the time series for artificial unit of time strictly related to the consecutive experiment stages.

The traditional approach to the hysteresis modeling assumes using differential equation models that involve the parameters that are specific to the modeled material as: Jiles–Atherton [9], Ylinen [10], Takács model [11], Prandtl–Ishlinskii model [12]. In most cases, the models are in the form of piecewise functions different for the particular branches of the hysteresis [10], [13].

In addition, the soft methods were considered: neural networks in the form of multi-layer perceptions, learned by the back propagation algorithm for supervised training [14], [15], or the Levenberg-Marquardt algorithm [16]–[18]. For considered experiments, successful modeling using supervised artificial neural networks learned by Kalman filter was already made [19].

MacKay in [20] suggested that Gaussian processes might be a replacement tool for supervised neural networks.

During numerical experiments, the influence of parameters of GP was examined. It was stated that the parameters of GP models are much more significant for the proper time series modeling then the parameters of ANN. The number of neurons, the initial values of ANN, values of the parameters that govern the learning process does not influence the numerical results much. The improper parameters of GP lead to incorrect modeling. Using GA is the well-known technique for supporting the process of ANN learning process [2], [21]–[24].

The aim of this survey was to confirm or subvert the thesis of the possibility of using GP instead of ANN for modeling hysteresis loops of stress-strain relation for concrete specimens and to find methodologies for effective selection of GP parameters. The tool, selected for this purpose was GA, and each individual in the population represents a possible solution of the Gaussian model, similarly to [18], but in this paper scatter, crossover operator was used.

3. Gaussian Process Model

Lets consider the stochastic process Y, generated by the set of fixed basis functions with random weights [3]:

$$Y(x) = \sum_{j=1}^{M} W_j \phi_j(x), \tag{1}$$

where x is the input vector indexing random variables [3]. If weights vector has normal distribution with zero mean, and particular standard deviation $W_j \sim N(0,\Sigma)$ then $E_W[Y(x)] = 0$ and $E_W[Y(x)Y'(x)] = \phi^T(x)\Sigma\phi(x)$. The training data set consist of pairs (x_i,t_i), where t_i, $i = 1, 2, \ldots, N$ is the sample from the random variable $T(x_i)$.
To make the prediction in the new input x_* it is necessary to compute conditional distribution $p(T(x_*)|T(x_1),\ldots,T(x_N))$. Let C denote covariance matrix of the training data, $t = [T(x_1), \ldots, T(x_N)]$, k denote the covariances vector between the training data $T(x_1),\ldots,T(x_N)$ and $T(x_*)$, V denote the prior variance of $T(x_*)$ that is $Cov(T(x_*),T(x_*))$. Then [3]:

$$E(T(x_*)|T(x_1),\ldots,T(x_N)) = k^T C^{-1} t, \tag{2}$$

$$D^2(T(x_*)|T(x_1),\ldots,T(x_N)) = V - k^T C^{-1} k. \tag{3}$$

Two covariance functions were considered:

- Squared exponential:

$$C(x^i,x^j) = v_0 \exp\Big(\sum_{l=1}^{d} a_l (x_l^i - x_l^j)^2\Big) + b, \tag{4}$$

where $x^i,x^j \in R^d$, $x^i = [x_1^i, \ldots, x_d^i]$. Then target covariance is given by:

$$C(x^i,x^j) + \sigma_v^2 \delta_{i,j}, \tag{5}$$

where

$$\sigma_v^2 \tag{6}$$

is the variance for the $p(T(x_*)|T(x_1),\ldots,T(x_N))$ $\delta_{i,j} = 1$ for $i = j$, $\delta_{i,j} = 0$ for i not equal j. The parameters of the model ale considered in the log space:

$$\theta = (\ln(v_0),\ln(b),\ln(a_1),\ldots,\ln(a_d),\ln(\sigma_v^2),\ln(\nu)). \tag{7}$$

- Rational quadratic:

$$C(x^i,x^j) = v_0\Big(1 + \sum_{l=1}^{d} a_l (x_l^i - x_l^j)^2\Big)^{-\tau} + b, \tag{8}$$

$$\theta = \big(\ln(v_0), \ln(b), \ln(a_1), \ldots, \ln(a_d), \ln(\sigma_v^2)\big). \tag{9}$$

Gaussian process structure may be viewed in the ANN form:

$$\theta = \big(\ln(v_0), \ln(b), \mathbf{w}, \ln(\nu)\big), \tag{10}$$

where b is the network bias, σ_v^2 is the noise incorporated into the network, $\mathbf{w}$ is the vector of weights, v_0 is the scaling parameter, x^i,x^j are input vectors [25]. That is why the effectiveness of GP models was compared to ANN models.

3.1. Learning Hyper Parameters

The non-linear optimizer is used to find the maximum likelihood values of the parameters θ_i. It is done by equaling to zero partial derivatives of log likelihood and using one of the standard optimization algorithms. The Scaled Conjugate Gradient optimization (SCG) was used with the standard Matlab implementation options [26]. The analysis showed that the key role in the efficiency of the procedure plays the number of SCG steps marked with k parameter.

The starting parameter values for the algorithm are randomly chosen from the $N(m,\sigma)$ distribution, where m and σ are the hyper parameters for this model. Additional noise term is added to the noise σ_v^2 in Eqs. (5) and (6) to make sure that noise variance never collapse to zero.

3.2. Making Predictions

During this stage the parameters for the predicted Gaussian distribution are computed according to the Eqs. (11) and (12) [3]:

$$E(T(x_*)|T(x_1),\ldots,T(x_N)) = k^T C^{-1} t, \tag{11}$$

$$D^2(T(x_*)|T(x_1),\ldots,T(x_N)) = V - C^T C^{-1} k. \tag{12}$$

3.3. Parameters of the Presented Procedure

The starting parameter values for the algorithm m and *sigma* and the SCG algorithm steps k number have to be chosen before each algorithm run. In the paper, the research for this parameters is made. Software for Flexible Bayesian Modeling and Markov Chain Sampling implementation [27] with own author's modification was used to implement the theoretical model. The weight number in each Gaussian model corresponds fully to the feed forward ANNs of the architecture that were considered in [19]. Each ANN weight has its equivalent in GP model.
To summarize the data effectiveness simulation set using one single value, *MSE* error was introduced:

$$MSEV = \frac{1}{V}\sum_{l=1}^{V}(y_l - \overline{y}_l)^2, \tag{13}$$

where: $V = L$, T is the number of learning and testing patterns, respectively; y_l – the target $\overline{y}_l$ is computed output mean value for l-th pattern scaled to the interval $[0\ldots1]$, see Eq. (1).

3.3.1. Calibrating the Parameters of the Numerical Models

As far as squared exponential covariance function is concerned the parameters of models are:

$$\theta = (\ln(v_0),\ln(b),\ln(a_1),\ldots,\ln(a_d),\ln(\sigma_v^2),\ln(\nu)). \tag{14}$$

For the rational quadratic covariance:

$$\theta = (\ln(v_0),\ln(b),\ln(a_1),\ldots,\ln(a_d),\ln(\sigma_v^2)). \tag{15}$$

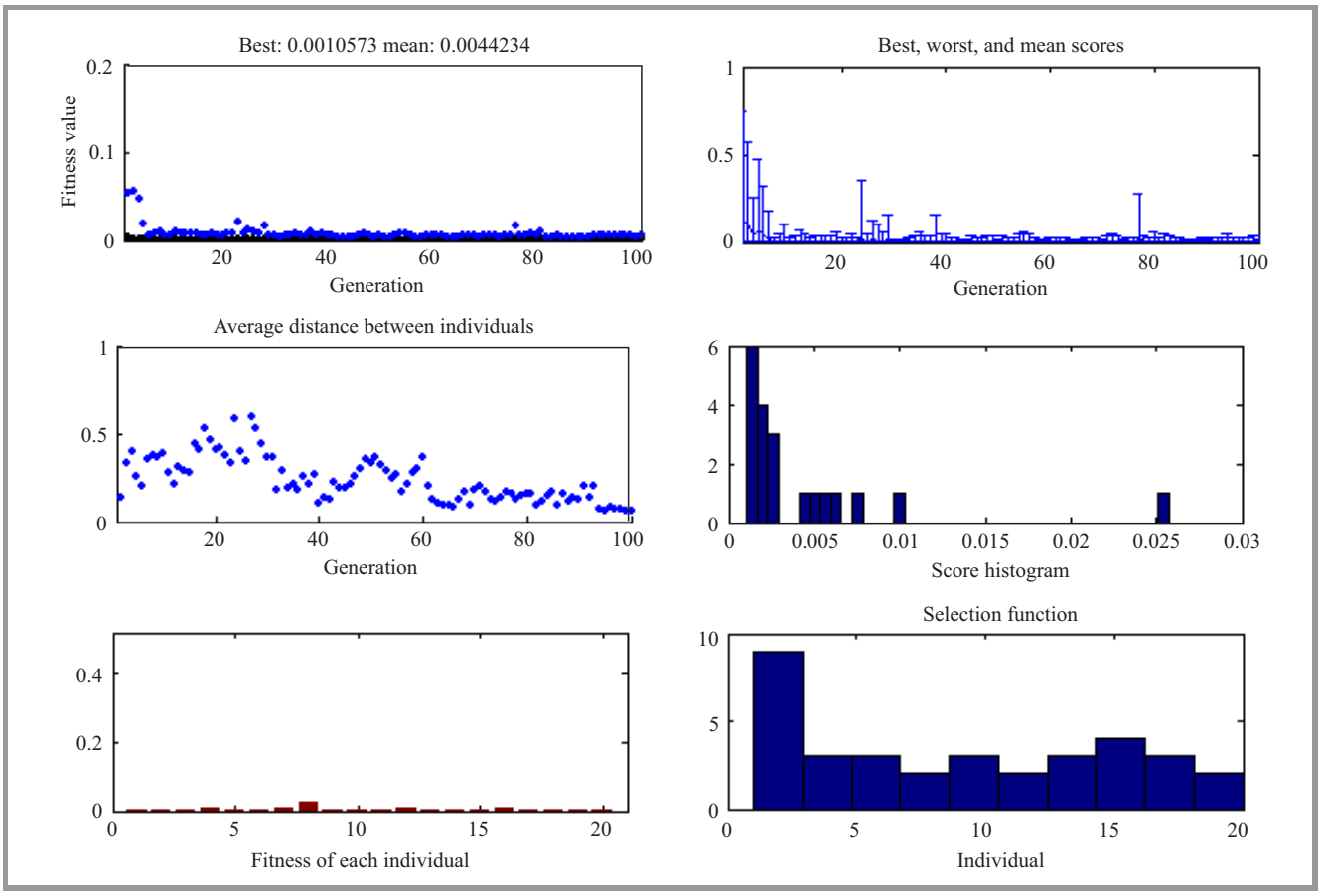

Fig. 1. Sample results of genetic algorithm proceeding.

To initialize the process of finding the optimal values of the parameters mean and variance of their prior Gaussian densities have to be set:

$$\theta(i) \sim N(m, \sigma). \tag{16}$$

3.3.2. The Genetic Algorithm for the GP Models Parameter Finding

There was stated that there is the significant influence of the parameters on the value of *MSE* errors as far as parameters (m, σ) and the length of SCG process k is concerned. This was the main reason for applying the additional procedure of calibrating both models. To reach that GA was chosen, and the fitness function to minimize was set in the form:

$$MSE(\overline{k}, m, \sigma), \tag{17}$$

where $\overline{k}$ is k scaled to the selected range by the linear scaling parameter ω: $\overline{k} = \omega k$.

In this research, the Matlab Genetic Algorithm Tool was used [28]. Population type, which specifies the type of the input to the fitness function was the vector $(\overline{k}, m, \sigma) \in R^3$. Population size, which specifies how many individuals there are in each generation, was assumed 20 individuals. The uniform creation function creates the initial population from the given interval of initial range. The scaling function, which converts raw fitness scores returned by the fitness function to values in a range that is suitable for the selection function was used. Next, the rank scaling function was applied. Rank scales the raw scores and is based on the rank of each individual, rather than its score. The rank of an individual is its position in the sorted scores. The rank of the fittest individual is 1, the next fittest is 2 and so on. Rank fitness scaling removes the effect of the spread of the raw scores [29].

The selection function chooses parents for the next generation based on their scaled values from the fitness scaling function.

Stochastic uniform selection function was then applied. It lays out a line in which each parent corresponds to a section of the line of length proportional to its expectation. The algorithm moves along the line in equal size steps, one step for each parent. At each step, the algorithm allocates a parent from the section it lands on. The first step is a uniform random number less than the step size. Reproduction options determine how the genetic algorithm creates children at each new generation. Elite count specifies the number of individuals that are guaranteed to survive to the next generation. Set elite count to be a positive integer less than or equal to population size. Here this number was set to 2. Then crossover fraction specifies the fraction of the next generation, other than elite individuals, that

are produced by crossover. The remaining individuals, other than elite individuals, in the next generation are produced by mutation. Set crossover fraction is a fraction between 0 and 1, and was set 0.8.

Mutation functions make small random changes in the individuals in the population, which provide genetic diversity and enable the GA to search a broader space. In presented research a Gaussian mutation functions was used. It adds a random number to each vector entry of an individual. This random number is taken from a Gaussian distribution centered on zero. The variance of this distribution can be controlled with two parameters. The scale parameter determines the variance at the first generation, and the shrink parameter controls how variance shrinks as generations go by. If the shrink is 0, the variance is constant. If the shrink is 1, the variance lowers to 0 linearly as the last generation is reached. Scale and shrink parameters was set at 1.

Crossover combines two individuals, or parents, to form a new individual, or child, for the next generation. Scattered crossover was used. It creates a random binary vector. It then selects the genes where the vector is a 1 from the first parent, and the genes where the vector is a 0 from the second parent, and combines the genes to form the child [29].

Stopping criteria determine what causes the algorithm to terminate:

- generations parameter specifies the maximum number of iterations the genetic algorithm performs (the value 100 was set),
- time limit specifies the maximum time (in seconds) the genetic algorithm runs before stopping (inf. was set),
- fitness limit – if the best fitness value is less than or equal to the value of fitness limit, the algorithm stops (inf. was set),
- stall generations – if there is no improvement in the best fitness value for the number of generations specified by stall generations, the algorithm stops (50 was set,
- stall time limit – if there is no improvement in the best fitness value for an interval of time in seconds specified by stall time limit, the algorithm stops (10^5 was set).

Best fitness plots the best function value in each generation vs. iteration number. Score diversity plots a histogram of the scores at each generation. Best individual plots the vector entries of the individual with the best fitness function value in each generation. Scores plots the scores of the individuals at each generation. Distance plots the average distance between individuals at each generation. Range plots the minimum, maximum, and mean fitness function values in each generation, see Fig. 1.

4. Simulation

Tests were made according to previously, investigated data sets [19]. Many experimental data sets coming from different kind of loading-unloading concrete and steel specimens were considered. These time series reflects the behavior of the material over time.

In this Section the time series coming from 12 concrete cylindric samples 3×6 inches size, that were compressed according to the following cyclic loading plan are presented [26]:

- monotonic increasing of the load to the maximal value,
- decreasing of the load to the 0,
- monotonic increasing of the load to the maximal value, and the stress-strain relation in time was consider as the modeled time series.

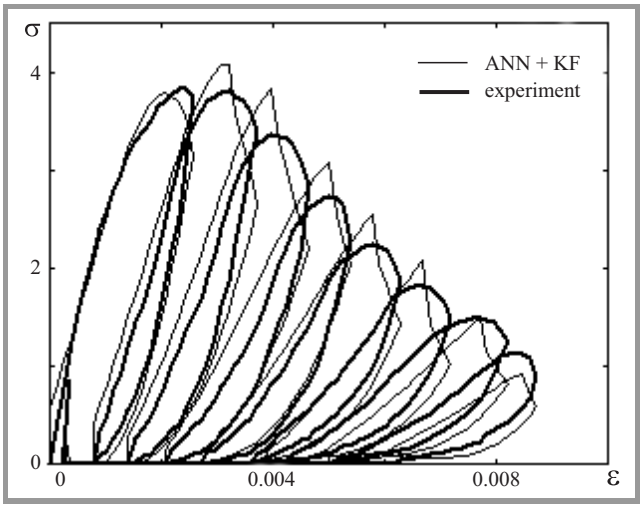

Fig. 2. Simulation and prediction of hysteresis loops from ANN models, learned by Kalman filtering (vertical axis contains values of stress, horizontal axis shows values of strain).

Data for calibrating and testing the models were discrete points from stress–strain σ-ε relation, see Fig. 2. As a preprocessing scaling to the internal range $[0.1 \ldots 0.9]$ was applied. This operation was done to correspond to the learning and testing for ANN, considered in [19] for the same experiment. Given data sets were divided into calibrating and testing set, corresponding to the learning set and testing set in [19]. The used testing set consists of points from 3 last of 8 hysteresis loops. For properly calibrated models it results in simulating behavior of the material in the first experiment part and prediction of the material behavior in the final part of the experiment basis on the material behavior in the first part of the experiment. The number of weight in each Gaussian model corresponds feed forward ANN of the architecture, which were considered in [19]. Each ANN weight has its equivalent in Gaussian process model. The output Gaussian process models

were the stress σ value, predicted in the current step of computation.
The calibrating set for the experiment consist of first six hysteresis loops, which gave $L = 273$ data points. The testing set were selected as the following three loops, what resulted in $T = 132$ data points. From among many input vectors the most effective

$$\mathbf{x}(j) = [\sigma(j-1), j/(273+132), marker_2],$$

was found in [19], where i is the number of current pattern, $j = 1, \ldots, 405$. The *marker* is the parameter numbering patterns for network learning and testing inside each loop separately independently from another loops. Parameters $marker_1$ and $marker_2$ were adopted. Inside i-th hysteresis loop the following values of these parameters were the most numerically effective:

$$marker_{1,i} = \Big[1/M_i, 2/M_i, \ldots, M_i/M_i, (M_i-1)/M_i, (M_i-2)/M_i, \ldots, (M_i-N_i)/M_i\Big] \quad (18)$$

where M_i is the number of experimental points for which the material is loaded, N_i is the number of experimental points which material is unloaded inside i-th hysteresis loop. Parameter $marker_{2,i}$ is based on $marker_{1,i}$ scaled to the interval $[0.1 \ldots 0.9]$.
There were the sets of parameters when both models are simulating and predicting the data set correctly. The combination of algorithm parameters was for example $m = 0$, $\sigma = 1$, $k = 10$.
Results of simulation given data set is presented for both considered covariance models initialized by the same set of parameters $m = 5, \sigma = 1, k = 40$. The numerical accuracy of both models differs, see Fig. 3. The predicted mean values of the experimental data for the rational quadratic covariance function are shown on the Fig. 4.

5. Results Discussion

The analysis of the presented results suggest that the k parameter setting plays the key role in the presented numerical method. For the chosen values of k both covariance function can be used to simulate presented data correctly. Setting the SCG algorithm steps maximal number too small or two large can make one of the covariance function model ineffective. Simulation made shows that the values of k in the range $[10 \ldots 45]$. For the squared exponential covariance function much longer SCG operating phase is necessary.
For the rational quadratic covariance function $k = 10$ is enough to receive correct results, see Figs. 4 and 5. For the squared exponential covariance model processing SCG algorithm for too less steps result in averaging the obtained results. Neither learning nor testing set is simulated correctly – obtained model do not distinguish the process taking place in time as far experiment is taking place. Then,

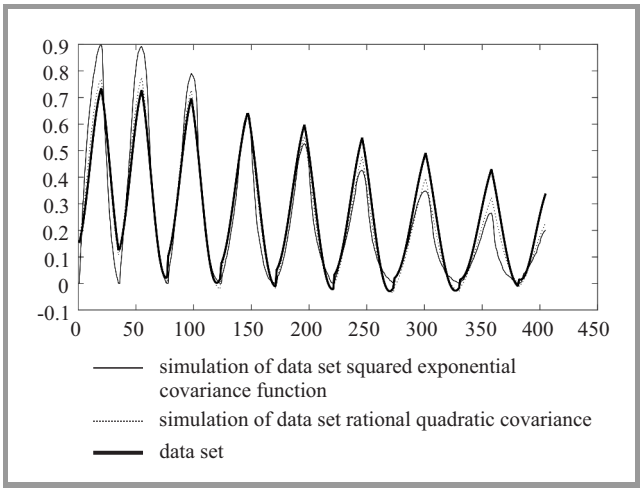

Fig. 3. Simulation and prediction of hysteresis loops for GP models (vertical axis presents values of stress, horizontal axis is number of data point j).

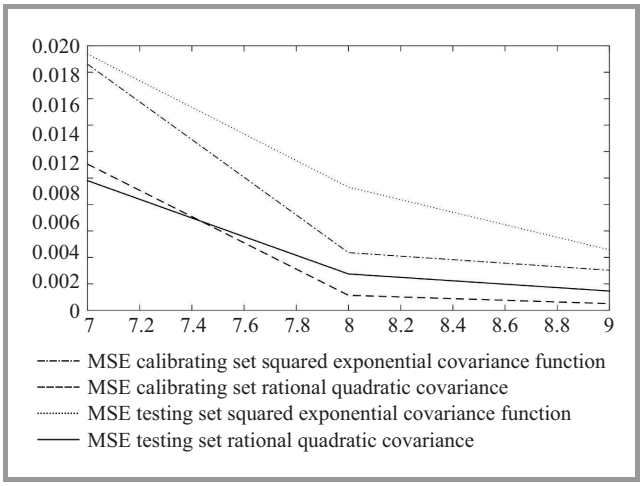

Fig. 4. MSE errors for calibration and prediction of hysteresis loops for GP models.

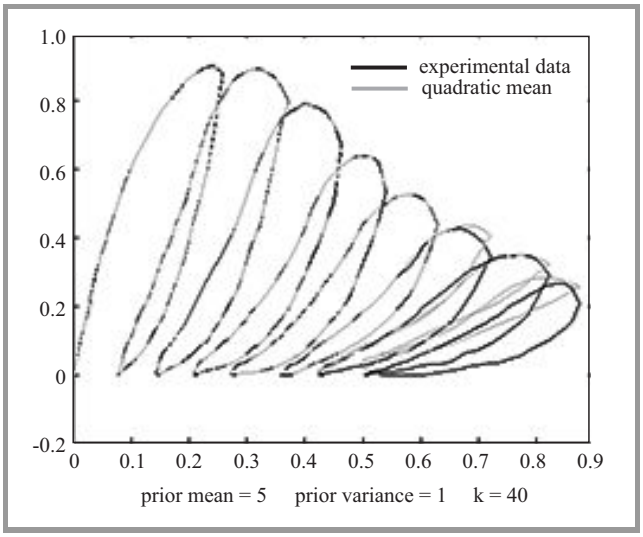

Fig. 5. Simulation and prediction of hysteresis loops for the rational quadratic covariance function GP models (vertical axis shows values of stress, horizontal axis presents values of strain).

setting k larger gives optimal behavior of the model. Processing SCG algorithm for too much steps result in averaging the obtained results again. The rational quadratic covariance function seems to be less sensitive to the k parameter setting. The effect of averaging results is taking place for considered range of $k > 70$ steps.
The comparison with earlier results when ANN learned by Kalman Filtering demonstrated superiority of Gaussian processes, as far as the quality of modeling is concerned, see Fig. 2. The model of first 7 hysteresis loops is more accurate. Last loop is model with less precision but the tendency still may be found.

6. Final Remarks

Gaussian processes were found as a very accurate tool for simulation and prediction of concrete hysteresis loops. The use of genetic algorithm as a method for automatic setting parameters of GP occurred to shorten the parameters setting process much. The simulation and prediction of the stress-strain relation is much precise then made by neural networks models.

References

[1] J. D. Hamilton *Time Series Analysis*. Princeton University Press, 1994.

[2] S. O. Haykin, *Neural Networks and Learning Machines*, 3rd ed. Prentice Hall, 2008.

[3] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[4] Y. Kwon, K. Kim, K. J. Tompkin, J. H. Kim, and C. Theobalt, "Efficient learning of image super-resolution and compression artifact removal with semi-local Gaussian processes", *IEEE Trans. Pattern Anal. and Machine Intell.*, vol. PP, no. 99, 2014.

[5] J. M. Young, "Probabilistic prediction of Alzheimer's disease from multimodal image data with Gaussian processes", Ph.D. thesis, University College London Press, 2015.

[6] A. K. Manshada and H. Rostamib, "Prediction of Wax precipitation in crude oil systems using Gaussian processes", *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects*, vol. 37, no. 1, pp. 84–91, 2015.

[7] J. Nielsen and J. Larsen, "Perception-based personalization of hearing aids using Gaussian processes and active learning", *IEEE/ACM Trans. Speech, and Lang. Process.*, vol. 23, no. 1, pp. 162–173, 2015.

[8] J. Diwale, S. Sanjay, L. Ioannis, and J. Colin, "Optimization of an airborne wind energy system using constrained Gaussian processes with transient measurements", in *Proc. Indian Control Conf. ICC 2015*, Chennai (Madras), India, 2015.

[9] M. Hamimid, S. M. Mimoune, and M. Feliachi, "Minor hysteresis loops model based on exponential parameters scaling of the modified Jiles–Atherton model", *Physica B: Condensed Matter*, vol. 407, no. 13, pp. 2438–2441, 2012.

[10] A. Ganczarski and L. Barwacz, "Low cycle fatigue based on unilateral damage evolution", *Int. J. of Damage Mechanics*, vol. 16, no. 2, pp. 159–177, 2007.

[11] K. Chwastek, "Modelling hysteresis loops in thick steel sheet with the dynamic Takács model", *Physica B: Condensed Matter*, vol. 407, no. 17, pp. 3632–3634, 2012.

[12] M. Al Janaideh, "A time-dependent stop operator for modeling a class of singular hysteresis loops in a piezoceramic actuator", *Physica B: Condensed Matter*, vol. 413, pp. 100–104, 2013.

[13] A. P. S. Baghel, A. Gupta, K. Chwastek, and S. V. Kulkarni, "Comprehensive modelling of dynamic hysteresis loops in the rolling and transverse directions for transformer laminations", *Physica B: Condensed Matter*, vol. 462, pp. 86–92, 2015.

[14] I. Kucuk, "Prediction of hysteresis loop in magnetic cores using neural network and genetic algorithm", *J. Magnetism and Magnetic Materials*, vol. 305, no. 2, pp. 423–427, 2006.

[15] R. Dong, Y. Tan, H. Chen, and Y. Xie, "A neural networks based model for rate-dependent hysteresis for piezoceramic actuators", *Sensors and Actuators A: Physical*, vol. 143, no. 2, pp. 370–376, 2008.

[16] A. Nouicer, E. Nouicer, and F. Mouloudc, "A neural network for incorporating the thermal effect on the magnetic hysteresis of the 3F3 material using the Jiles–Atherton model", *J. Magnetism and Magnetic Materials*, vol. 373, pp. 240–243, 2015.

[17] V. Wolfs and P. Willems, "Development of discharge-stage curves affected by hysteresis using time varying models, model trees and neural networks", *Environ. Model. & Softw.*, vol. 55, pp. 107–119, 2014.

[18] X. Zhang, Y. Tan, and M. Su, "Modeling of hysteresis in piezoelectric actuators using neural networks", *Mechan. Syst. and Sig. Process.*, vol. 23, no. 8, pp. 2699–2711, 2009.

[19] A. Krok, "Analiza wybranych zagadnień mechaniki konstrukcji i materiałów za pomocą SSN i filtrów Kalmana (Analysis of mechanics of structures and material problems applying artifcial neural networks learnt by means of Kalman filtering)", Ph.D. thesis, Tadeusz Kościuszko Cracow University of Technology, 2007 (in Polish).

[20] D. J. C. MacKay, "Gaussian processes – a replacement for supervised neural networks?", Lecture notes for a tutorial at Neural Information Processing Systems (NIPS) 1997, Cambridge University, 1997.

[21] D. Whitley, T. Starkweather, and C. Bogart, "Genetic algorithms and neural networks: optimizing connections and connectivity", *Parallel Comput.*, vol. 14, no. 3, pp. 347–361, 1990.

[22] D. Pham and D. Karaboga, *Intelligent Optimisation Techniques: Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks*. Springer, 2011.

[23] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[24] Ms. Dharmistha and D. Vishwakarma, "Genetic algorithm based weights optimization of artificial neural network", *Int. J. Adv. Res. Elec., Electron. and Instrumen. Engin.*, vol. 1, no. 3, 2012.

[25] R. M. Neal, "Regression and classification using Gaussian process priors", in *Bayesian Statistics 6*, J. M. Bernardo *et al.*, Eds. Oxford University Press, 1998, pp. 475–501.

[26] B. P. Sinha, K. H. Gerstle, and L. G. Tulin, "Stress-strain relations for concrete under cyclic loading", *J. of the American Concrete Institute*, no. 61-12, 1964.

[27] R. Neal, Software for Flexible Bayesian Modeling and Markov Chain Sampling [Online]. Available: http://http://www.cs.toronto.edu/~radford/fbm.software.html

[28] Optimization Toolbox User's Guide – MathWorks, The MathWorks Inc. [Online]. Available: http://uk.mathworks.com/help/pdf-doc/optim/optim_tb.pdf

[29] S. N. Sivanandam and S. N. Deepa, *Introduction to Genetic Algorithms*. Springer, 2008.

[30] Y. S. Othmana *et al.*, "Frequency based hysteresis compensation for piezoelectric tube scanner using Artificial Neural Networks", *Procedia Engin.*, vol. 41, pp. 757–763, 2012.

[31] F. Pernkopf and D. Bouchaffra, "Genetic-Based EM Algorithm for Learning Gaussian Mixture Models", *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 27, no. 8, pp. 1344–1348, 2005.

Agnieszka Krok – for biography, see this issue, p. 51.